Easy Data Protection with the Azure DevOps Backup Tool

Data protection and backup are critical considerations for any organization leveraging **Azure DevOps** for their software development and project management needs. **The Azure DevOps Backup Tool** offers a comprehensive solution that simplifies the backup and restoration of Azure DevOps resources, ensuring the safety and availability of an organization's data. In this blog post, we will explore the features, benefits, and usage of the Azure DevOps Backup Tool.

- Check out the Azure DevOps Backup Tool on **Azure DevOps Marketplace**: https://marketplace.visualstudio.com/items?itemName=solidify.ado-backup-tool
- Visit https://solidify.dev/ and check out or other products, or get in touch with us for expert consultant services.

Main feature overview

Azure DevOps Backup Tool enables you to:

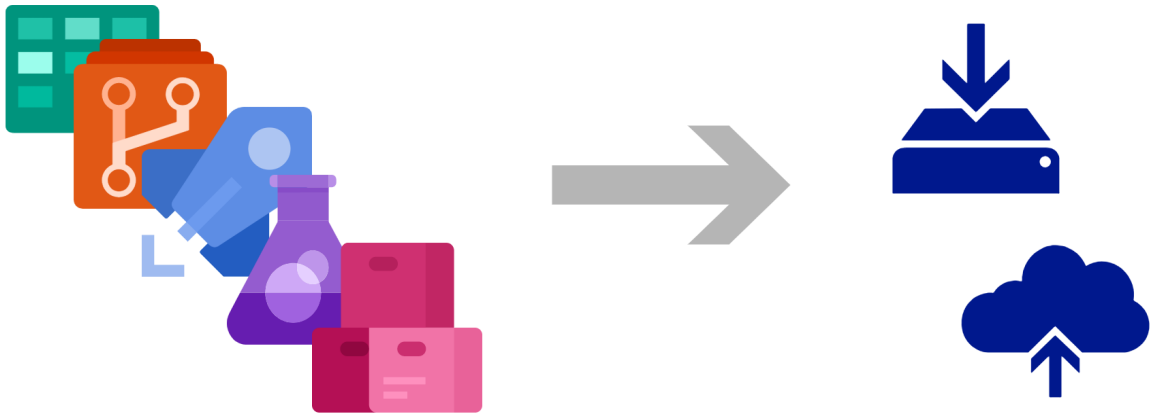
- Create backups of your Azure DevOps resources in the form of JSON files stored in a local directory or network file share. You may then restore these backups to your production environment at any time. Our tool has support for all of the following data sources inside Azure DevOps:
  - Azure Boards
  - Azure Repos
  - Azure Pipelines
  - Azure Test Plans
  - Azure Artifacts
- Easy Integration into **Azure DevOps Pipelines**.
- **Seamlessly perform data migrations** between Azure DevOps projects, or consolidate data from multiple sources into a single Azure DevOps project.
  - Supports both Azure DevOps Services organizations and Azure DevOps Server colletions.

Flexible Backup Options

The tool allows you to back up specific Azure DevOps resources based on your requirements. No matter what subset of your resources, and no matter whether you want to back up a single project or an entire organization, the Azure DevOps Backup Tool offers flexibility in selecting the scope of your backups.

Comprehensive Resource Support

The Azure DevOps Backup Tool supports a wide range of Azure DevOps resources, ensuring that your critical data is fully protected. This includes areas such as git repositories, pipelines, work items, work item process models, test plans, artifacts and much more.



Why You Should Consider the Azure DevOps Backup Tool

Azure DevOps Backup Tool acts as an insurance policy that will help you ensure business continuity, allowing your developers to continue working seamlessly and with minimal downtime even in the face of unforeseen incidents.

1. Data Recovery and Permanent Deletion Protection

Azure DevOps provides the option to permanently delete resources, and this can pose risks if important data is accidentally removed or your company falls victim to a cyberattack. The Azure DevOps Backup Tool ensures that you can recover important data even in the event of accidental deletion or a security breach.

- By default, Azure DevOps resources are usually soft deleted for 30 or 90 days depending on the resource, and can be recovered through the REST API during the soft delete window, but cannot be recovered after that.
- With Azure DevOps Backup Tool you can retain your data indefinitely.

2. Efficient and Targeted Backups

Unlike traditional backup methods that involve backing up and restoring the entire Azure DevOps database, the Azure DevOps Backup Tool offers a more streamlined approach. With this tool, you can easily and selectively back up specific resources, eliminating the need for lengthy downtime and minimizing the impact on your development process.

3. Compliance and Regulatory Requirements

Many enterprises are subject to auditing and regulatory requirements that necessitate robust data protection measures. By having a comprehensive backup strategy in place, you can ensure data integrity and demonstrate compliance with regulatory standards.

Features

- Backup and restore Azure DevOps resources locally (Single project or whole organization)
- Migrate or or consolidate Azure DevOps resources between Azure DevOps instances and projects

Task overview

Azure DevOps Backup Tool provides a set of pipeline tasks that backs up Azure DevOps data as json files to a local file directory.

The extension consist of the following tasks:

- **Azure DevOps Backup Tool: Export** - Run export jobs via a pipeline task. Save the downloaded files to a local file area or network file share.
- **Azure DevOps Backup Tool: Import** - Run import jobs via a pipeline task. Read the files from a local file area or network file share.

Supported Resources

The following resources are supported as of today (2023-05), with more and more being added continually:

- Area/Iteration paths
- Teams and team settings
    - Team board configuration
- Git repositories
    - Pull requests
    - Branch policies
- Project wikis
- Pipelines
    - Build definitions
    - Release definitions
    - Task groups
    - Variable groups
    - Environments (Deployment groups)
- Work Items
    - Revision history
    - Comments
    - Attachments
    - Links
        - Work Items links
        - Code links (repo/commit/branch/tag)
- Work Item Queries
- Dashboards
- Work Item Process models
    - Inherited
    - XML-based
- Testplans (no test data)
- Artifacts
    - Feeds and packages

- NuGet
- Npm
- Universal
- Python (export only)

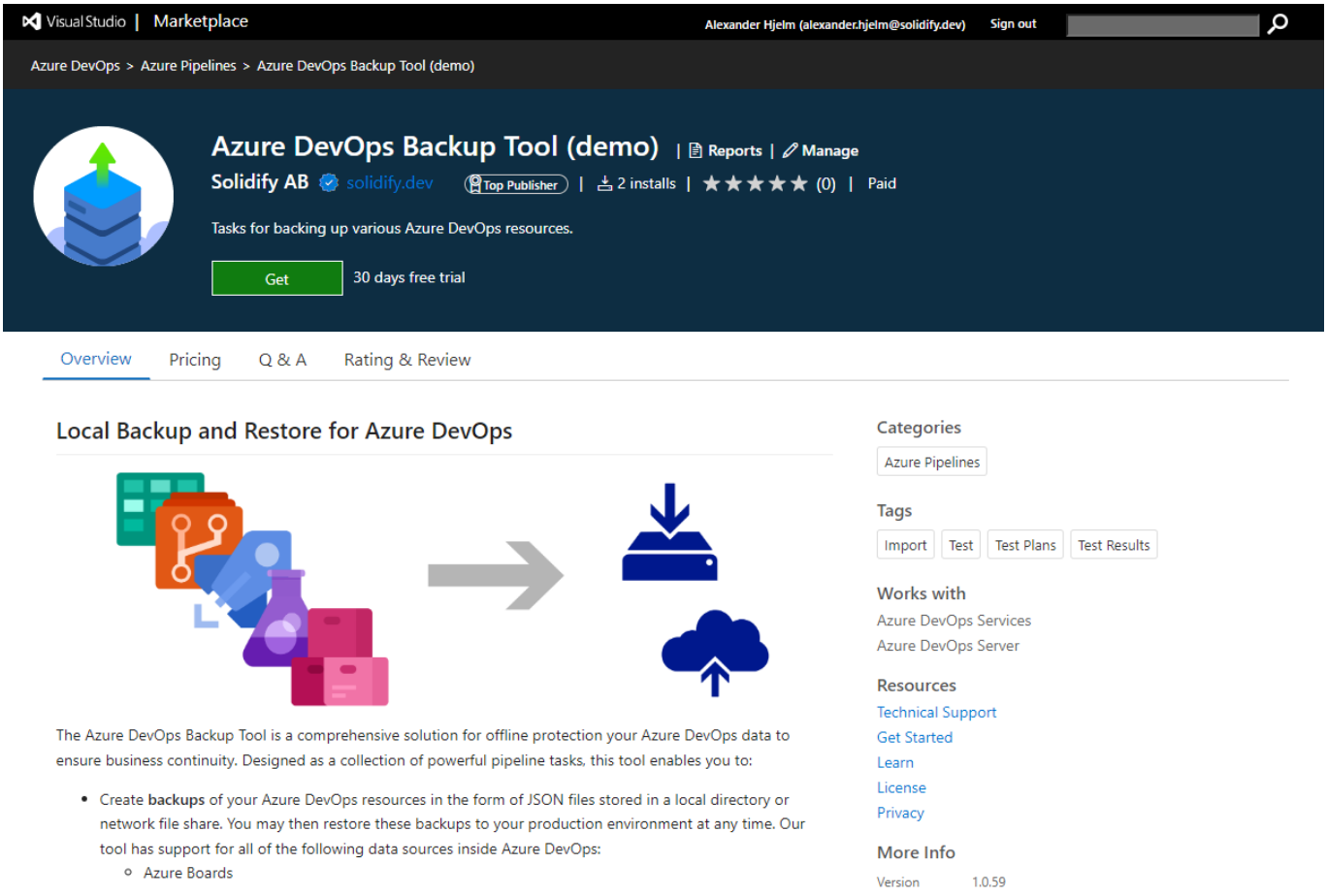
How to use

Let us walk through the steps to set up and use the Azure DevOps Backup Tool effectively.

1: **Download the Extension**

Start by downloading the Azure DevOps Backup Tool extension from the Azure DevOps Marketplace. This extension provides the necessary tasks:

https://marketplace.visualstudio.com/items?itemName=solidify.ado-backup-tool



2a: Set Up a **Self-Hosted Build Server** (Optional)

If you prefer to use a self-hosted build server, follow these steps:

- Set up a Windows Server 2016+ environment.
- Install the following dependencies:
  - .NET Core 7.0 Runtime
  - git
  - pip
  - gradle
  - Azure CLI (az)

- Ensure that the build server has access to the internet and Azure DevOps.
- Set up the build agent.

2b: Use a **Microsoft Hosted Windows Build Agent**

Alternatively, you can use a Microsoft-hosted Windows build agent, which comes pre-configured with all the necessary prerequisites. This simplifies the setup process and ensures compatibility.

3: Set Up a Dedicated Backup Server and Expose the Disk

Prepare a backup server and expose the disk as a network file share. This server will be used to store the backup copies of your Azure DevOps resources. Ensure that the build server has access to this network file share.

The recommended setup is as follows:

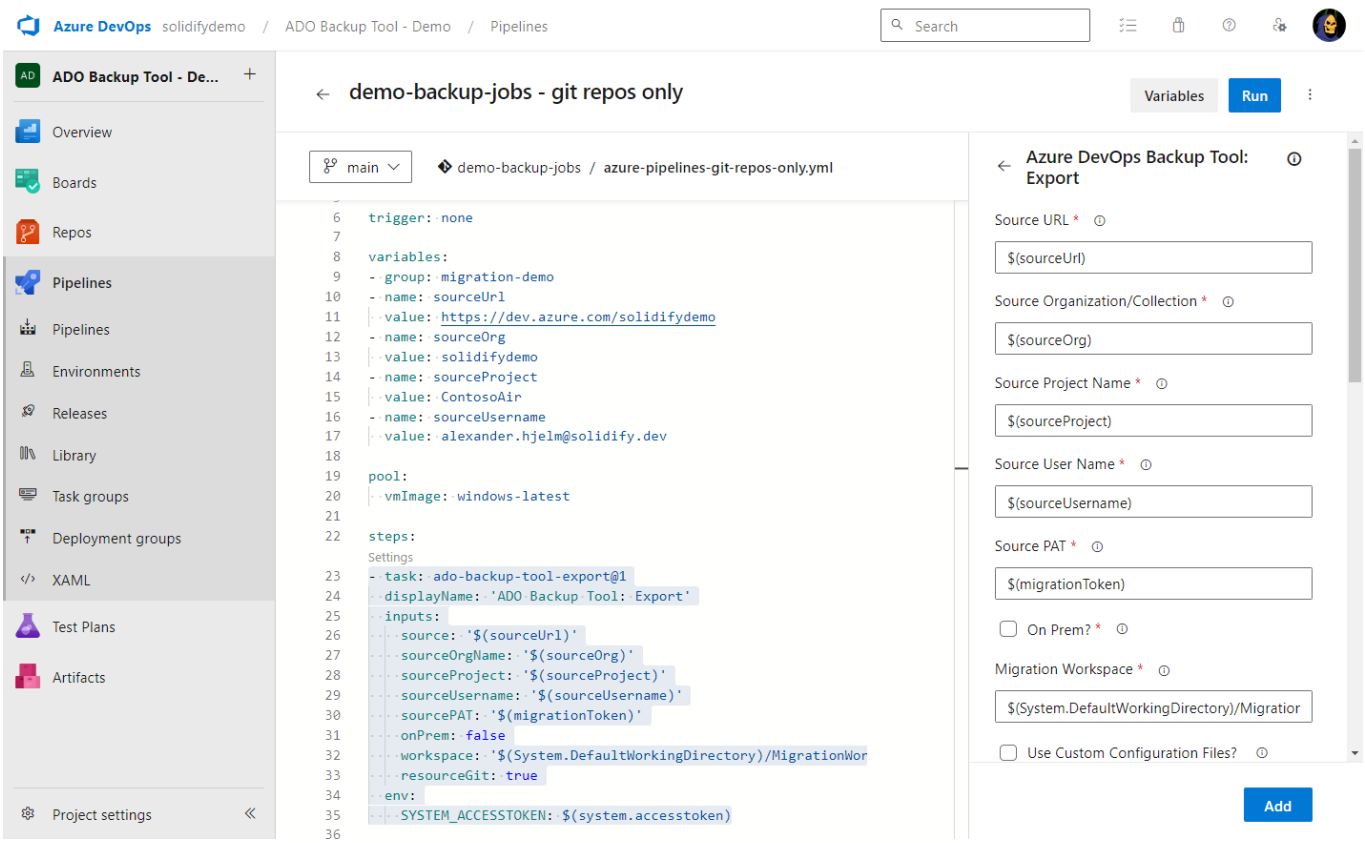
1. Expose the drive to the **ADO Pipeline Agent VM** as a network share.
2. Mount the network share as a drive on the **ADO Pipeline Agent VM**.
3. Finally, in the Export task, specify the **Migration Workspace** as a path on the mounted network drive.

Here is an example server architecture:

Server	Platform	Firewall opening
(1) Azure DevOps Server	Windows Server	(2)
(2) Pipeline Agent	Windows Server	(1) and (3)
(3) Dedicated Backup Server	Any	(2) (via network share)

4: Configure the Backup Pipeline with Azure Pipelines

Create an Azure Pipeline to set up the backup process. Use the Export Task provided by the Azure DevOps Backup Tool extension to export your desired Azure DevOps resources. Additionally, use the FileCopy task to copy the exported resources to your network file share on the backup server.



Here is an example yaml pipeline that publishes the exported resources as a build artifact for debug and demo purposes:

```yaml
trigger: none

variables:
- group: migration-demo
- name: sourceUrl
  value: https://dev.azure.com/solidifydemo
- name: sourceOrg
  value: solidifydemo
- name: sourceProject
  value: ContosoAir
- name: sourceUsername
  value: alexander.hjelm@solidify.dev

pool:
  vmImage: windows-latest

steps:
- task: ado-backup-tool-export@1
  displayName: 'ADO Backup Tool: Export'
  inputs:
    source: '$(sourceUrl)'
    sourceOrgName: '$(sourceOrg)'
    sourceProject: '$(sourceProject)'
    sourceUsername: '$(sourceUsername)'
    sourcePAT: '$(migrationToken)'
    onPrem: false
```

```
      workspace: '$(System.DefaultWorkingDirectory)/MigrationWorkspace'
      resourceGit: true
    env:
      SYSTEM_ACCESSTOKEN: $(system.accesstoken)

- task: ArchiveFiles@2
  inputs:
    rootFolderOrFile: '$(System.DefaultWorkingDirectory)/MigrationWorkspace'
    includeRootFolder: false
    archiveType: 'zip'
    archiveFile: '$(Build.ArtifactStagingDirectory)\$(Build.BuildNumber).zip'
    replaceExistingArchive: false

- task: PublishPipelineArtifact@1
  inputs:
    targetPath: '$(Build.ArtifactStagingDirectory)\$(Build.BuildNumber).zip'
    artifact: 'Export'
    publishLocation: 'pipeline'
```

5: Configure the Scheduling of the Export Task

To automate the backup process, configure the scheduling of the Export Task within your Azure Pipeline. Choose a suitable schedule, such as running the backup daily at midnight or weekly.

Here is an example schedule:

```
schedules:
- cron: "0 0 * * *"
  displayName: Nightly Build
```

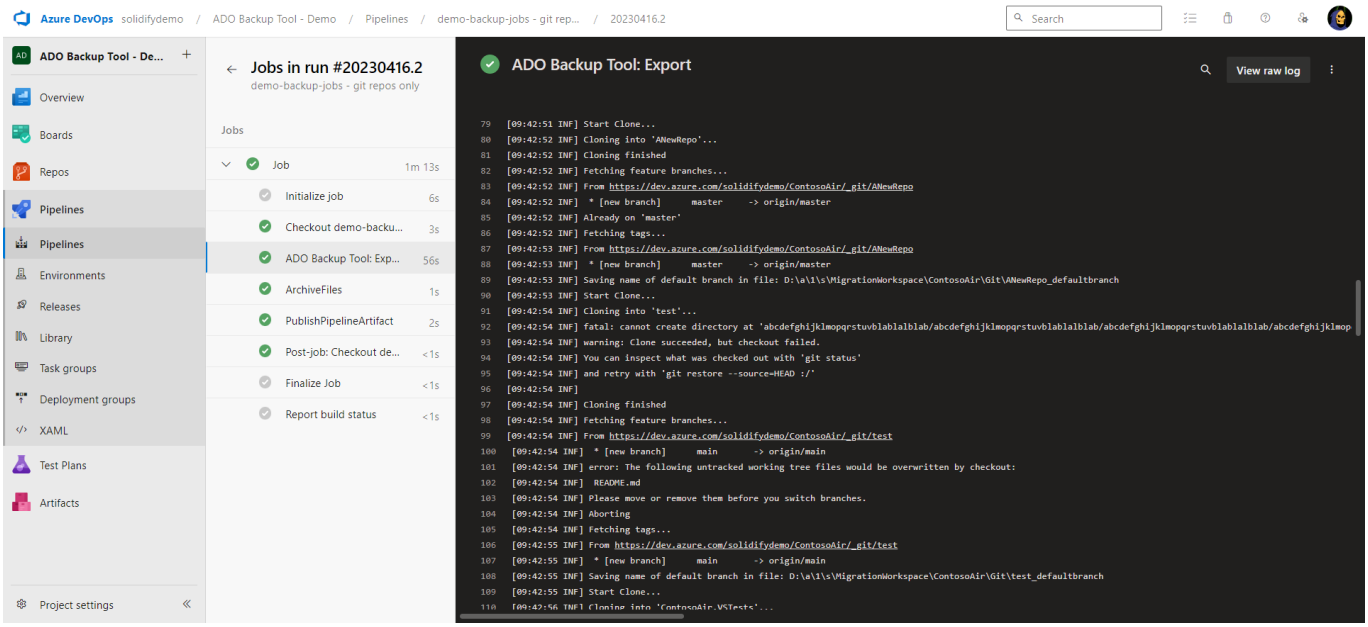
6: Grant Correct Permissions to the Build Service Account

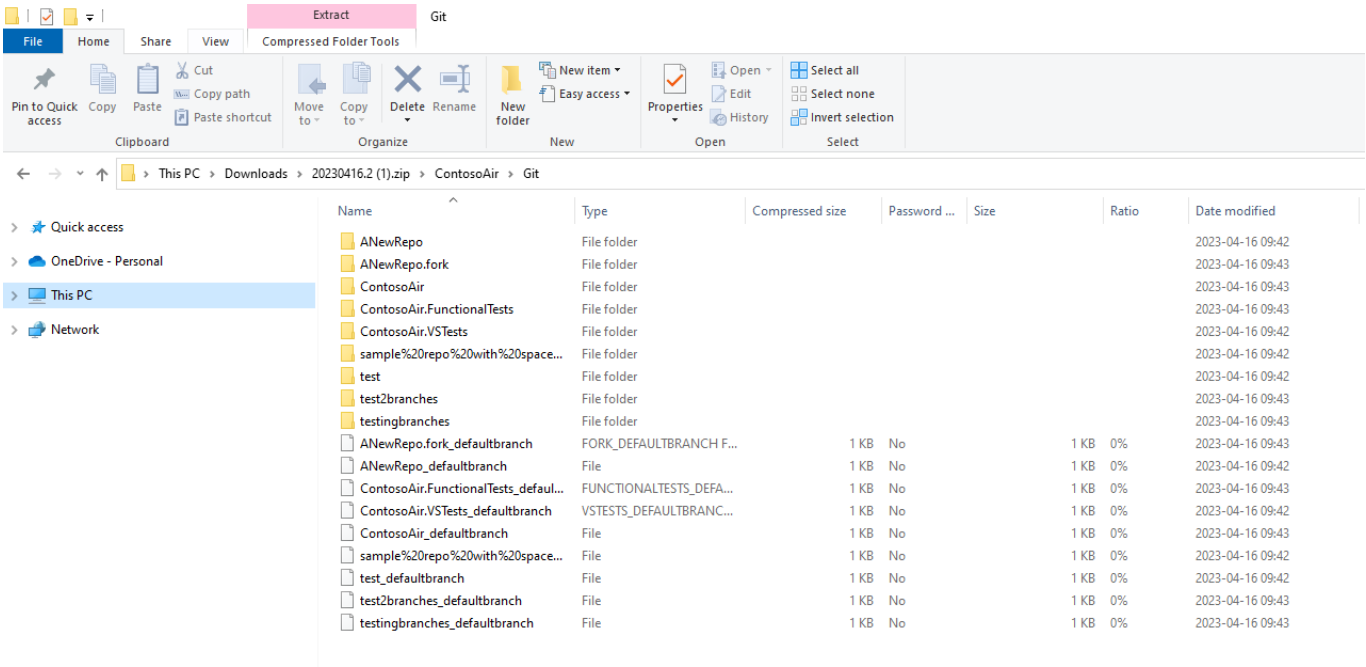
Ensure that the build service account has the necessary permissions in the target Azure DevOps project. At minimum, the account should have the following role:

- **Reader** (for Export operations only)
- **Contributor** (for Export+Import operations)

7: Run the Backup Pipeline

Trigger the backup pipeline and monitor its execution. Verify that all your desired Azure DevOps resources are successfully exported and written to the backup server as expected.





8: Restore Data Using the Import Task

To restore your backed-up data, set up the Import Task within an Azure Pipeline. Manually trigger the pipeline and use the FileCopy task to copy the resources from the backup server to your pipeline workspace. Finally, run the Import Task to restore the resources to your target Azure DevOps environment.

Here is an example yaml pipeline for triggering a resource import:

```yaml
trigger: none

variables:
- group: migration-demo
- name: sourceUrl
  value: https://dev.azure.com/solidifydemo
- name: sourceOrg
  value: solidifydemo
```

```
  - name: sourceProject
    value: ContosoAir
  - name: sourceUsername
    value: alexander.hjelm@solidify.dev

  pool:
    vmImage: windows-latest

  steps:
  - task: CopyFiles@2
    inputs:
      SourceFolder: 'X:\Backup\ContosAir'
      Contents: '**'
      TargetFolder: '$(System.DefaultWorkingDirectory)/MigrationWorkspace'

  - task: ado-backup-tool-import@1
    inputs:
      target: '$(sourceUrl)'
      sourceOrgName: '$(sourceOrg)'
      targetOrgName: '$(sourceOrg)'
      sourceProject: '$(sourceProject)'
      targetProject: '$(sourceProject)'
      targetUsername: '$(sourceUsername)'
      targetPAT: '$(migrationToken)'
      onPrem: false
      workspace: '$(System.DefaultWorkingDirectory)/MigrationWorkspace'
      resourceGit: true
```

Conclusion

Azure DevOps Backup Tool enables organizations to safeguard their Azure DevOps data, streamline backup and restoration processes, and facilitate efficient migration across instances and projects. Using the tool will ensure business continuity and reduce the risk of data loss.

Start using the **Azure DevOps Backup Tool** today and take control of your Azure DevOps data!

- Check out the Azure DevOps Backup Tool on **Azure DevOps Marketplace**: https://marketplace.visualstudio.com/items?itemName=solidify.ado-backup-tool
- Or visit us at https://solidify.dev/